



CONTROLLER SERVICES

- › DebuggerService
- › ModulesService
- › InterceptorService
- › ExceptionService
- › PluginService
- › HandlerService
- › RequestService

SAMPLE HANDLER CODE

```
component{  
  
    any function index(event){  
        return "Hello";  
    }  
  
}
```

CORE PLUGINS

- › AntiSamy
- › ApplicationStorage
- › BeanFactory
- › CFCViewer
- › ClientStorage
- › ClusterStorage
- › CookieStorage
- › DateUtils
- › FeedGenerator
- › FeedReader
- › FileUtils
- › i18n
- › IOC
- › JavaLoader
- › JSON
- › JVMUtils
- › Logger
- › MailService
- › MessageBox
- › MethodInjector
- › ORMService
- › QueryHelper
- › Renderer
- › ResourceBundle
- › SessionStorage
- › StringBuffer
- › Timer
- › Utilities
- › Validator
- › Webservices
- › XMLConverter
- › Zip

CORE INTERCEPTORS

- › Autowire
- › ColdboxSideBar
- › Deploy
- › EnvironmentControl
- › Security
- › SES

HANDLER PROPERTIES

- › This scope properties
- › event_cache_suffix
- › preHandler_only
- › preHandler_except
- › postHandler_only
- › postHandler_except
- › allowedMethods

IMPLICIT HANDLER EVENTS

- › preHandler,pre{Action}
- › postHandler,post{Action}
- › aroundHandler,around{Action}
- › onMissingAction

FOLDER CONVENTIONS

- › Application.cfc
- › index.cfm (empty)
- › config
 - › Coldbox.cfc
 - › Routes.cfm
- › handlers
- › model
- › modules
- › layouts
- › views

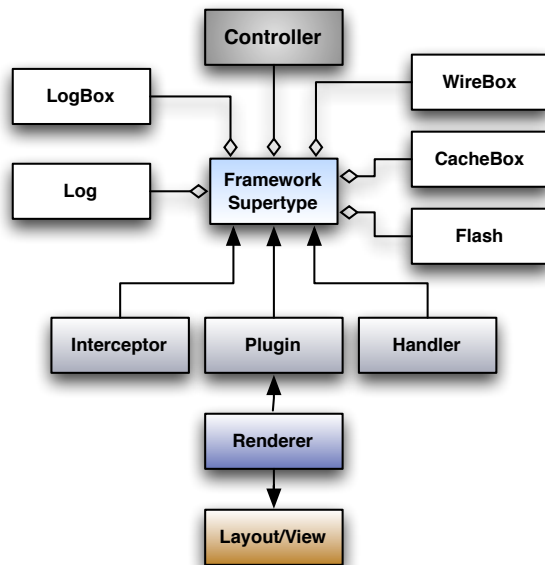
FRAMEWORKSUPERTYPE

- › \$abort(), \$dump(), \$htmlhead(), \$include(), \$throw(), \$rethrow()
- › addAsset(assetList)
- › announceInterception(state,[interceptData])
- › getColdBoxOCM([cachename])
- › getController()
- › getDatasource(alias)
- › getDebugMode()
- › getfwLocale()
- › getInterceptor(name)
- › getModel(name,[dsl],[initArguments])
- › getMailSettings()
- › getMyPlugin(plugin,[newInstance])
- › getPlugin(plugin,[customPlugin],[newInstance])
- › getResource(resource,[default],[locale])
- › getSetting(name,[fwSetting])
- › getSettingsBean()
- › getSettingStructure([fwSetting],[DeepCopyFlag])
- › includeUDF(udf.library)
- › persistVariables(persist,[persistStruct])
- › populateModel(model,scope,trustedSetter,include,exclude)
- › renderExternalView(view)
- › renderLayout(layout)
- › renderView([view],[cache],[Timeout],[LastAccessTimeout])
- › runEvent(event,[prepostExempt],[private])
- › setDebugMode(mode)
- › setNextEvent(event,[queryString],[addToken],[persist],[persistStruct],[ssl],[baseURL],[postProcessExempt],[URL],[URI],[statusCode])
- › setSetting(name,value)
- › setFWLocale(locale)
- › settingExists(name,[fwSetting])

CONTROLLER COMMON METHODS

- › getAppRootPath()
- › getCacheBox()
- › getLogBox()
- › getWireBox()
- › getColdboxOCM([name])
- › getColdboxSettings()
- › getConfigSettings()
- › getPlugin(plugin,[customPlugin],[newInstance])
- › get{ServiceName}()
- › getSettingStructure([fwSetting],[DeepCopyFlag])
- › getSetting(name,[fwSetting])
- › persistVariables(persist,[persistStruct])
- › runEvent(event,[prepostExempt],[private])
- › settingExists(name,[fwSetting])
- › setSetting(name,value)
- › setNextEvent(event,[queryString],[addToken],[persist],[persistStruct],[ssl],[baseURL],[postProcessExempt],[URL],[URI],[statusCode])

COLDBOX COMMON CLASSES



INTERCEPTOR METHODS

- › configure()
- › getProperties(properties)
- › setProperties()
- › getProperty(name)
- › setProperty(name,value)
- › propertyExists(name)
- › unregister(state)
- › clearBuffer()
- › appendToBuffer(string)
- › getBufferString()
- › getBufferObject()

REQUEST CONTEXT COMMON METHODS (RECEIVED AS EVENT TO EVENT HANDLERS)

- › buildLink(linkto,[translate],[ssl],[baseURL],[queryString])
- › clearCollection()
- › collectionAppend(collection,[overwrite])
- › getCollection([DeepCopyFlag],[private])
- › getCurrentAction()
- › getCurrentEvent()
- › getCurrentHandler()
- › getCurrentLayout()
- › getCurrentModule()
- › getCurrentRoute()
- › getCurrentRoutedURL()
- › getCurrentView()
- › getDefaultLayout()
- › getDefaultView()
- › getEventName()
- › getHTTPHeader(header,[default])
- › getHTTPMethod()
- › getModuleRoot()
- › getSESBaseURL(), setSESBaseURL()
- › getSize()
- › getTrimValue(name,[default],[private]), getValue(name,[default],[private])
- › isAjax()
- › isProxyRequest()
- › isSES()
- › isSSL()
- › noExecution()
- › noRender([remove])
- › overrideEvent(event)
- › paramValue(name,value,[private])
- › renderData(type,data,[contentType],[encoding],[statusCode],[statusText],[jsonCase],[jsonQueryFormat],[jsonAsText],[xmlColumnList],[xmlUseCDATA],[xmlListDelimiter],[xmlRootName])
- › removeValue(name,[private])
- › setHTTPHeader(statusCode,statusText,name,value,charset)
- › setLayout(name)
- › setValue(name,value,[private])
- › setView(name,[noLayout],[cache],[cacheTimeout],[cacheLastAccessTimeout],[cacheSuffix],[layout])
- › valueExists(name,[private])
- › showDebugPanel(boolean)



SAMPLE INTERCEPTOR

```
component{
    function configure(){
        //configure properties here
    }
    //method = interception point
    function preprocess(interceptData,
        event){
    }
}
```

SAMPLE PLUGIN

```
component{
    function init(){
        setPluginName("MyPlugin");
        setPluginVersion("1.0");
        setPluginDescription("MyPlugin");
        setPluginAuthor("YourName");
        setPluginAuthorURL("www.my.com");
    }
}
```

COLDBOX CONFIG

- › Coldbox (struct)
- › Settings (struct)
- › Conventions (struct)
- › Environments (struct)
- › loc (struct)
- › WireBox (struct)
- › Debugger (struct)
- › mailSettings (struct)
- › l18n (struct)
- › BugTracers (struct)
- › Webservices (struct)
- › Datasources (struct)
- › LayoutSettings (struct)
- › Layouts (array of structs)
- › Cachebox (struct)
- › Logbox (struct)
- › InterceptorSettings (struct)
- › Interceptors (array of structs)
- › Modules (struct)

CONFIG INJECTED VARS

- › AppMapping
- › Controller
- › LogBoxConfig

SES ROUTE METHODS

- › addRouter(*)
- › addModuleRoutes(pattern, module)
- › setBaseURL()
- › setUniqueURLS()
- › setAutoReload()
- › setExtensionDetection()
- › setValidExtensions()
- › setThrowOnInvalidExtension()
- › setLooseMatching()

* ADDROUTE()

- › [pattern]
- › [handler]
- › [action]
- › [matchVariables]
- › [view]
- › [viewNoLayout]
- › [valuePairTranslation]
- › [constraints]
- › [module]

SES ROUTES

- › Default Pattern :handler/:action?
- › :var ANY Placeholder
- › :var-numeric Numeric
- › :var-alpha Alpha
- › :var{X} Limit by X
- › ? = optional placeholder
- › :regex() Limit by regex
- › Valid Extensions: xml, json, json, html, htm, rss

URL ACTIONS (INDEX.CFM?)

- › fwreinit=1 or {reinitPassword}
- › debugmode=true
- › debugpass={DebugPass}
- › fwcache=anything

COLDBOX INTERCEPTION POINTS

- › afterConfigurationLoad()
- › afterAspectsLoad()
- › afterPluginCreation() *
 - › pluginPath, custom, oPlugin
- › afterHandlerCreation() *
 - › handlerPath, oHandler
- › afterModelCreation() *
 - › modelName, oModel
- › applicationStart()
- › applicationEnd()
- › onException() *
 - › exception
- › onReinit()
- › preprocess()
- › postProcess()
- › preEvent()
 - › processedEvent
- › preLayout()
- › preProxyResults()
- › preRender() *
 - › renderedContent
- › postEvent() *
 - › processedEvent
- › postRender()
- › preViewRender()
 - › View, cache, cacheTimeout, cacheLastAccessTimeout, cacheSuffix, module
- › postViewRender()
 - › View, cache, cacheTimeout, cacheLastAccessTimeout, cacheSuffix, module, renderedView
- › sessionStart()
- › sessionEnd()

* Has intercept data

CACHEBOX INTERCEPTION POINTS

- › afterCacheElementInsert()
 - › Cache, cacheObject, cacheObjectKey, cacheObjectTimeout, cacheObjectLastAccessTimeout
- › afterCacheElementRemoved()
 - › Cache, cacheObjectKey
- › afterCacheElementExpired()
 - › Cache, cacheObjectKey
- › afterCacheElementUpdated()
 - › Cache, cacheNewObject, cacheOldObject
- › afterCacheClearAll()
 - › Cache
- › afterCacheRegistration()
 - › Cache
- › afterCacheRemoval()
 - › Cache (name)
- › beforeCacheRemoval()
 - › Cache
- › beforeCacheReplacement()
 - › oldCache, newCache
- › afterCacheFactoryConfiguration()
 - › cacheFactory
- › beforeCacheFactoryShutdown()
 - › cacheFactory
- › afterCacheFactoryShutdown()
 - › cacheFactory
- › beforeCacheShutdown()
 - › cache
- › afterCacheShutdown()
 - › Cache

* Has intercept data

SES INTERCEPTOR PROPERTIES

- › configFile : relative or absolute routes file defaults to (config/Routes.cfm)

ENVIRONMENTCONTROL INTERCEPTOR PROPERTIES

- › configFile : relative or absolute file location
- › fireOnInit : [boolean=true]

SECURITY INTERCEPTOR PROPERTY

- › useRegex : [boolean=true]
- › rulesSource : [string=xml, db, ioc, ocm, model]
- › queryChecks : [boolean=true]
- › preEventSecurity : [boolean=false]
- › validator : [classpath]
- › validatorIOC : [bean name]
- › validatorModel : [model name]

AUTOWIRE INTERCEPTOR PROPERTIES

- › completeDIMethodName : [string=onDIComplete]
- › enableSetterInjection : [boolean=false]
- › annotationCheck : [boolean=false]
- › entityInjection : [boolean=false]
- › entityInclude : [string]
- › entityExclude : [string]

DEPLOY INTERCEPTOR PROPERTIES

- › tagFile : relative or absolute file location
- › deployCommandObject : instantiation path
- › deployCommandModel : [model name]

COLDBOXSIDEBAR INTERCEPTOR PROPERTIES

- › baseAppPath : [string=cgi.script.name]
- › cssPath : [string]
- › jsPath : [string]
- › URLParamNameList : [string]
- › Yoffset : [numeric]
- › slideSpeed : [numeric]
- › waitTimeBeforeOpen : [numeric]
- › waitTimeBeforeClose : [numeric]
- › isScroll : [boolean]
- › Links : [array]
- › Width : [numeric]
- › visibleWidth : [numeric]
- › imagePath : [path]
- › imageVAlign : [numeric]



WireBox

WIREBOX CUSTOM DSL INTERFACE

```
Interface{
    any init(injector){}
    any process(definition){}
}
```

WIREBOX PROVIDER INTERFACE

```
Interface{
    any get(){}
}
```

WIREBOX CUSTOM SCOPE INTERFACE

```
Interface{
    any init(injector){}
    any getFromScope(mapping,
        initArguments){}
}
```

WIREBOX PROVIDER METHODS

```
function getXXX() provider="Mapping"{
}
```

WIREBOX BINDER CONFIG

- › logboxConfig (path)
- › cacheBox (struct)
- › scopeRegistration (struct)
- › customDSL (struct)
- › customScopes (struct)
- › scanLocations (array)
- › stopRecursions (array)
- › parentInjector (instance)
- › listeners (array of structs)

WIREBOX SCOPES

- › NOSCOPE OR PROTOTYPE
- › SINGLETON
- › CACHEBOX
- › SESSION
- › APPLICATION
- › REQUEST
- › SERVER

WIREBOX BUILD TYPES

- › CFC
- › JAVA
- › WEBSERVICE
- › RSS
- › DSL
- › CONSTANT
- › FACTORY
- › PROVIDER

CORE INJECTION DSL

- › Id or model
- › Id:{name}
- › Id:{name}:{method}
- › Provider
- › Provider:{name}
- › Cachebox
- › Cachebox:{cache}
- › Cachebox:{cache}:{key}
- › Logbox
- › Logbox:root
- › Logbox:logger:{category}

COLDBOX INJECTION DSL

- › Coldbox
- › Coldbox:setting:{setting}
- › Coldbox:fwSetting:{setting}
- › Coldbox:plugin:{plugin}
- › Coldbox:myplugin:{myPlugin}
- › Coldbox:datasource:{alias}
- › Coldbox:configBean
- › Coldbox:fwConfigBean
- › Coldbox:interceptor:{name}
- › Coldbox:loaderService
- › Coldbox:debuggerService
- › Coldbox:handlerservice
- › Coldbox:interceptorservice
- › Coldbox:moduleservice
- › Coldbox:requestservice
- › Coldbox:pluginservice
- › entityService
- › entityService:{entity}
- › loc
- › loc:{beanName}
- › Javaloader:{class}
- › Webservice:{alias}

WIREBOX OBJECT LIFE CYCLE EVENTS

- › afterInjectorConfiguration()
 - › injector
- › beforeInstanceCreation()
 - › mapping
- › afterInstanceCreation()
 - › Mapping, target
- › afterInstanceInitialized()
 - › Mapping, target
- › beforeInstanceInspection()
 - › Mapping, binder
- › afterInstanceInspection()
 - › Mapping, binder

INJECTOR COMMON METHODS

- › Autowire(target,[mapping],[targetID],[annotationCheck])
- › clearSingletons()
- › containsInstance(name)
- › getBinder()
- › getCacheBox()
- › getColdBox()
- › getLogBox()
- › getEventManager()
- › getInstance([name],[dsl],[initArguments])
- › getObjectPopulator()
- › getParent()
- › getScope(scope)
- › getScopes()
- › getVersion()
- › locateInstance()
- › setParent(injector)

OBJECT PERSISTENCE ANNOTATIONS

- › singleton
- › scope=[scopeName]
- › cache - Cache in CacheBox's default provider
- › cacheBox=[provider]
- › cacheTimeout = minutes
- › cacheLastAccessTimeout = minutes

PROPERTY, SETTER, CONSTRUCTOR ARGUMENT ANNOTATIONS

- › Inject=DSL
- › Provider="Mapping"

METHOD ANNOTATIONS

- › Inject=DSL
- › Provider="Mapping"
- › onDICComplete

INJECTOR-COLDBOX AWARENESS

If an object has the following methods, wirebox will call them for you with the right dependency:

- › setBeanFactory(injector)
- › setInjector(injector)
- › setColdBox(coldbox)

MAPPING DSL INITIATORS

- › map("name")
- › mapPath("path")
- › mapDirectory("instanation path")
- › with("name")

MAPPING DSL DESTINATIONS

- › to("CFC")
- › toJava("java class path")
- › toDSL("dsl string")
- › toFactoryMethod(mapping, method)
- › toRSS("rss URL")
- › toValue(constant value)
- › toWebservice("URI")
- › toProvider("provider mapping name")

MAPPING DSL SCOPE PERSISTENCE

- › asSingleton()
- › inCacheBox(key,timeout,lastAccessTimeout,provider)
- › Into(scope)

MAPPING DSL CONSTRUCTOR ARGUMENT METHODS

- › initWith(arguments) - init with specific arguments
- › initArg(name, ref, dsl, value, javaCast)

MAPPING DSL SETTER/PROPERTY METHODS

- › Property(name, ref, dsl, value, javaCast, scope)
- › Setter(name, ref, dsl, value, javaCast)

MAPPING DSL FACTORY OBJECTS

```
map("name")
    .toFactoryMethod(mapping, method)
    .methodArg(name, ref, dsl, value, javaCast) // Repeat as needed
```

REGISITER CUSTOM DSL AND SCOPES

Register custom DSL and Scopes via mapping DSL

- › mapDSL(namespace, path)
- › mapScope(scope,path)

MAPPING DSL MISCELLANEOUS

- › asEagerInit() - Not lazy loaded
- › constructor(method) - Override the default constructor of 'init'
- › noAutowire() - Don't autowire object
- › noInit() - Do not call constructor on object
- › onDICComplete(methods) - Register an array of methods for post processing
- › cacheBox(configFile,cacheFactory,enabled,classNamespace)
- › listener(class, properties, name)
- › logBoxConfig(filePath)
- › parentInjector(injector)
- › removeScanLocations(locations)
- › scanLocations(locations)
- › scopeRegistration(enabled,scope,key)
- › stopRecursions(recursionPaths)